微信指令系统的研发与应用

摘　要：新媒体时代，如何有效提升微信的阅读量与转发量，本文通过萧山日报技术中心自主研发的微信指令系统进行了探索与应用，利用微信提供的开发接口，结合 java+mysql 进行开发，实现了一套转发系统，并合理地进行制度设计，同比单条微信的转发量与阅读量提升接近 20%，通过技术手段有效地提升了新媒体的传播力度。

关键词：微信；阅读量；Java　Mysql；数据库连接池
中图分类号：G202　　文献标识码：A
文章编号：1671-0134（2018）02-075-03　　DOI：10.19483/j.cnki.11-4653/n.2018.02.028

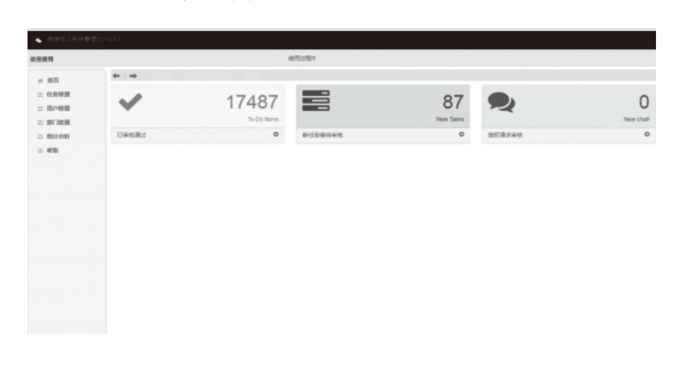
文 / 俞凌峰

1. 微信指令系统的背景与现状

随着新媒体的异军突起，微信已经越来越成为人们依赖的社交软件，腾讯公司推出的微信订阅号、服务号等也被越来越多的企业所采用。对媒体人而言，最常见的就是通过订阅号每天更新发布新闻，随着订阅号的增多，问题也慢慢出现。一是发布渠道单一。现代人关注的不仅仅是一个公众号，如果仅通过官方的发布，而不主动推送到人们眼前，很多人不会主动去寻找推送的微信；二是转发数低。转发数低就不能通过以点带面的方式形成传播的扩散，无法有效提高阅读量与点赞数。

基于这两点问题，可以通过采用单位员工集中转发的方式来提高微信图文消息的转发数与阅读量，但如何发送需要转发的微信消息（以下简称指令），并统计转发的指令成了一个问题。为了解决这个问题，萧报技术中心打算利用微信提供的公众平台开发工具 SDK，设计一套系统来完成指令的推送与统计等问题。

运行至今，共计发出微信指令约 700 余条，审核通过数 17000 余条，有效地提高了微信的转发数与阅读量，也调动了员工的转发积极性。另外，此系统也引起了行业同行的兴趣，我们将之复制并进行了一定的推广。



系统后台首页图



微信端截图

2. 系统需求与分析

通过调研，系统的流程：首先建立一个微信服务号——“萧报转发”，让单位的员工都关注这个服务号（需要有一道审核机制以排除非本单位人员），通过后台系统推送消息给用户，用户接收到消息后转发指令到朋友圈并截图，然后通过服务号上传截图，审核通过后系统将推送一条审核通过的消息给用户，并且系统需要对此做一个记录，以方便统计。

通过分析，判断转发系统需要实现以下几个模块：

用户管理：实现用户的关注申请以及审核。

指令管理：实现指令的编辑、推送、审核。超时的指令不能审核通过。

部门管理：实现部门的建立以及用户的分组。

统计分析：通过不同的方式，对用户的完成数进行统计，奖优罚劣。

权限设计：设计好权限的职责范围，将部分权限发放给二级管理员进行管理。

系统选型上，手机端采用微信提供的 weui 框架，后台通过目前比较成熟的 Tomcat+Java+Bootstrap+Mysql 搭配组合，完成后台系统的 UI 以及逻辑和数据库开发，服务器环境采用安全稳定的阿里云环境，实现云端部署。

3. 系统开发与实现

3.1 数据库的开发与连接池设计

通过对数据库的分析，建立 ER 模型，把表拆分为 in_customer 表（用户表）、in_datastatistic 表（数据统计表），in_department 表（部门分类表）、task 表（任务表）等七张表，其中，特别需要注意的是：由于微信的昵称现在引入了 emoji 表情，原先的存储引擎 utf8 已经不能满足新类型的字符串，因此，数据库以及表都需要将存储引擎改为 utf8mb4 引擎，这个新的引擎支持 emoji 表情的存储，适用范围更广泛。

其次，因为本套系统的指令发出以后，短时间内可能会接到较多的连接请求，为了提高并发量，采用了数据库连接池的设计方法，有助于提高对于数据库连接的效率和连接数量。

instructiondb
- in_customer
- in_datastatistic
- in_departments
- in_task
- in_tasksend
- in_user
- in_usertaskinfo

3.2 MVC 分层开发与优化

程序的逻辑代码层采用 java 语言开发，采用面向对象的设计方法进行类的拆分与设计，严格遵循了 MVC 的分层设计思想，主要的代码逻辑处理流程为：view 层→servlet 层（service）→数据库访问层→数据库连接池→数据库。通过分层以后，代码结构耦合性降低，复用性更高。

对于数据的访问层，将对数据的增删改查统一封装在 DbDao 层，方便对数据库进行常用操作进行统一管理，同时有助于对数据安全性的检查，提高系统的安全性。

为了以后很好地进行复制与扩展，将系统的数据库连接配置文件、微信配置文件，以及系统配置项文件进行了提取整合，当有新的项目需要复制时，直接进行对应的修改即可，可重复性大大提高。

实际使用过程中，对 tomcat 的生产环境进行优化调整，经实际测试，发现默认的 java 虚拟机 jvm 分配的内存不能满足实际生产环境的需要，系统有时候由于 jvm 设置的内存大小只有 256M，有时候内存溢出会引起死机等问题。经调整到 2G 以后，生产环境下内存使用率能够满足我们的实际需要。

JVM

Free memory: 647.39 MB Total memory: 1021.00 MB Max memory: 3641.00 MB

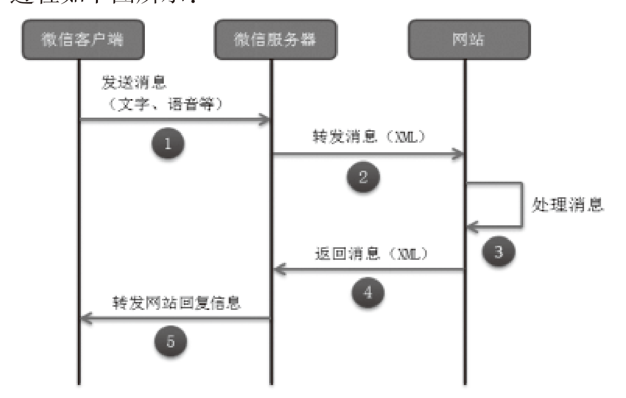
Memory Pool	Type	Initial	Total	Maximum	Used
PS Eden Space	Heap memory	256.50 MB	334.50 MB	1359.00 MB	168.43 MB (12%)
PS Old Gen	Heap memory	683.00 MB	683.00 MB	2730.50 MB	204.92 MB (7%)
PS Survivor Space	Heap memory	42.50 MB	3.50 MB	3.50 MB	0.25 MB (7%)
Code Cache	Non-heap memory	2.43 MB	4.25 MB	48.00 MB	4.06 MB (8%)
PS Perm Gen	Non-heap memory	512.00 MB	512.00 MB	2048.00 MB	55.78 MB (2%)

"ajp-apr-8009"

优化后的 JVM 内存使用率图

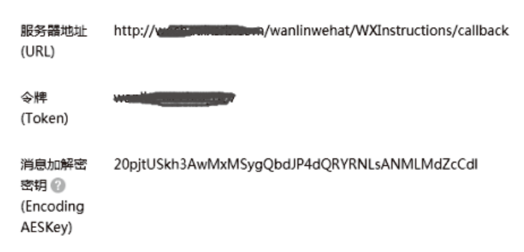
3.3 微信配置以及 access_token 定期刷新

由于是在微信的接口上进行开发，因此，必须遵循微信服务平台的一些约定。微信和阿里云服务器的通信过程如下图所示：



微信客户端与系统通信流程图

在微信平台网页端需要对接入的网站进行一些基本的配置，通过设置微信服务器转发消息到阿里云平台，并设置相应的服务器地址以及 Token。



公众平台开发配置图

由于 Access Token 有效期只有 7200 秒，而每天调用获取的次数只有 2000 次，所以，需要将 Access Token 进行缓存，保证不触发超过最大调用次数。另外，在微信公众平台中，绝大多数高级接口都需要 Access Token 授权才能进行调用，开发者需要使用中控服务器统一进行缓存与更新，以避免各自刷新而混乱。本系统中，Access Token 我们采用缓存的方式进行存储，避免由于单天转发数量过大而引起 Access Token 调用次数达到上限。

4. 制度配套

为了更好地调动广大员工的积极性进行微信转发，

（下转第 93 页）